REMARKS/ARGUMENTS

Reconsideration of this application is respectfully requested.

The Examiner's attention is directed to the fact that the last office action fails to address pending claims 70-72 (e.g., see pages 14-15 of the amendment submitted December 30, 2010).

The rejection of claims 39, 42-44, 48-52 and 54-69 under 35 U.S.C. §103 as allegedly being made "obvious" based on Smith '224 in view of Margulis '449 and Snape '922 is respectfully traversed.

In light of this new ground of rejection, applicants have elected to simply cancel all previously pending claims without prejudice or disclaimer in favor of a fresh new set of claims 73-92. Accordingly, the following remarks concerning the cited references will focus upon these new claims and especially upon new independent method claim 73 and system claim 92.

New claims 73-92 clarify that the first "gap" interval is imposed responsive to receipt of the global control message requiring gapping and without waiting for a further call to be received to trigger the gap interval. Subsequently, however, call gaps are imposed by waiting for calls to be received.

Smith teaches that a controller located at a network server calculates (responsive to an overload condition at the network server) a single computed value that consists of

- 10 -

the fraction of received traffic each traffic source may send on to that network server

"...a controller controls the number of transactions admitted to a network server..." The

output of a server overload controller is a computed value ("admission factor") repre-

senting the fraction of new transaction requests a source may send to the server during

the coming measurement interval" (see 5:41- 45 and 29-35). The single value that is

computed centrally by the network server overload controller of Smith consists of "a

fraction" that regulates how many of the transaction requests received should be

processed at the point they are received for forwarding to their destination (here the

network server). The way the "fraction" is calculated is then described in various

embodiments, in one of which Smith discusses how a "focused overload" that is a

sudden demand in increase for one service (see 8:56-57) is dealt with (in this case, the

overload controller may selectively target the service generating the focused overload).

Smith confirms that determining admission factors and implementing them are

separate, independent operations (see 12:65-67) and discusses adaptive gapping as a

means for sources to implement the admission factors that are generated by the

network server controller (see 12:63-67). In Smith, the central controller generates a

global "admission factor" that is a fractional value and all access servers must then

figure out how best to attain this fractional reduction using some sort of gapping regime.

Smith describes adaptive gapping as being performed by traffic sources each

time the controller tells the sources that its congestion level has changed (see 13:6-9).

The adapted gap interval is calculated using a predetermined equation in which the source measures its admission rate, and based on this and the admission factor, generates a new gap interval. The gap interval could be directly generated at the server. However, as Smith states, the server does not know and cannot directly measure the input transaction rate λ. If the network server (the central location) is to try to generate a future gap interval, then it does this by estimating the source's output transaction rate λ based on the present gap interval and using this to derive the new gap interval according to a given formula.

So, starting from Smith, a person of ordinary skill in the art is taught <u>away</u> from the applicants' claimed invention. For example, Smith teaches:

> i) the distribution of an admission factor C followed by a mechanism to either determine a local call gapping interval based on the admission factor C and the local source's admission rate λ; and as an alternative

> ii) that the central server must calculate the admission factor C and then guess the local source's admission rate λ to generate a central call gapping interval which is then not further modified locally.

- 12 -

Accordingly, if a person of ordinary skill in the art were to read Smith, they would realize:

> i) implementing a global overload control function gives rise to all sources which receive the same input transaction rate λ applying the same call gapping interval, which could result in repeated overload conditions at the network server if calls are re-admitted in high call volume rate bursts at the end of each gapping interval; and

> ii) if there are lots of traffic sources with low call rates then a long time may pass before these traffic sources apply the first call gap.

To address this problem in a system with multiple different types of traffic sources/network access points, the applicants' claimed invention firstly randomizes the size of the initial gap between zero and the respective local gap interval, and then ensures this is applied responsive to the global gapping constraint being received, i.e., without waiting for further traffic to be received. By immediately applying a random sized local gap interval, the overload condition at the network access controller is regulated rapidly without overly congesting the network access points as each gap is suitably scaled between no gap and the local gap interval and applied immediately. This is not taught or rendered obvious by the cited prior art.

Margulis teaches each switch (i.e., each network access point) in a network as monitoring its own load condition and when an overload condition is detected resulting from calls to a particular terminating number (a Mass Calling Event ("MCE")), this is reported by the switch to a network processor (i.e., a "network access controller").

In Margulis, the network processor then generates a gap control message to all of the switches of the network identifying the congested terminating number and a global gap time. All of the switches in the network then place a time-guard **between** calls based on the broadcast global gap time (see the abstract of Margulis, for example).

The key point is that it is "between calls". That is, a first call has to arrive to trigger the gap to be imposed. To avoid "synchronization" issues, Margulis teaches applying a random multiplier of between 0 and 1 to the **globally broadcast** initial gap constraint. This effectively off-sets the initial gap times applied by each switch so that network-wide call bursts at the end of each gap time are avoided. However, this might not be suitable for a network in which some access points have very much lower call rates such as that contemplated by the applicants' claimed invention.

Nothing in Margulis teaches applying an initial random gap interval without waiting for a call to be received by a switch. This is because in an MCE, calls are received at a very high rate by the majority of switches, so this is unlikely to be an issue in Margulis. In Margulis, different types of switches apply call gaps dependent on the

- 14 -

number of ineffective attempts (IAs) they receive. In Margulis, each access point has to perform a calculation based on the number of ineffective attempts (IAs) detected in the previous gap cycle.

There is nothing to prompt a person who is skilled in the art to consider how a switch with a very low call rate would respond to such a call gapping process. In Margulis, this is treated in subsequent cycles by the low number or lack of any IAs on such switches.

Moreover, a problem exists if the maximum initial gap interval that is applied according to Margulis is a random variant of the broadcast **global** gap interval. This global value, even if scaled randomly, may result in a first gapping interval totally unsuitable for some types of switches.

Snape describes how overload of a SSF is dealt with. In Snape, at 2:58-63, it is taught that "[i]f the SCF is in overload, or the service load is too high, it can send a Call-Gap instruction to the originating SSP to reduce the number of calls handled for that service..." Snape also describes how a SSP comprises a SSF and a call control function (CCF) (1:27-28) and that "known arrangements may allow some manual call-gapping at the CCF under control of an operator via the Man-Machine Interface..."

Col. 3, lines 10-29 of Snape then describes:

> "...Call-Gapping is executed at the point where the incoming
> call is recognized as an IN call, i.e., corresponding to where

- 15 -

>the call enters the IN...[e.g., the CCF]...when the SCP
>detects a traffic load in excess of a certain level it auto-
>matically issues an instruction to initiate a call-gap operation
>to the originating SSP...the SSP passes the instruction to the
>receiving CCF...for automatic implementation of Call-
>Gapping on the incoming traffic routes. Thus automatic Call-
>Gapping is implemented at the CCF. The more efficient
>implementation...causes execution of the Call-Gap operation
>in the CCF *before the call is passed to the SSF.*"

The Examiner is requested to reconsider his interpretation of Snape. Col. 2, lines 28-29, of Snape clearly indicates that the CCF is part of the SSF, which are both part of the SSP where a call is received. Accordingly, it is clear from Snape that a call must trigger the initial gap interval being applied by the SSP (the Service Switching Point), i.e., the call has in fact arrived at a respective network access point. However, within that SSP, Snape requires that the call has not yet been passed from the CCF to the SSF.

In other words, in Snape, the gap is applied only after a call is received, so a call does trigger the gap interval. However, the triggering call does not pass from the CCF to the SSF.

It is submitted that Snape, therefore, teaches a local access point applying a gap function that commences when a call is received during a condition of overload of a central network server.

None of the above prior art teaches or suggests imposing a randomized local gap interval directly responsive to an instruction to implement call gapping. In all of the cited prior art (and by technical convention), a call gapping process is a process applied "between" calls – whereas the claimed invention applies the first gap before any further traffic is received, i.e., without waiting for a call to be received to trigger the onset of the call gapping interval.

This is a simple, yet effective, way of enabling a central controller to regulate traffic arriving at an access point rapidly without relying on a call triggering a gapping interval. It addresses a problem that only becomes apparent when one considers that a network controller may be overloaded by a high number of access points (i.e., a large fan-in scenario) in which some calls are offered at what might be a fairly low rate for one or more of the access points. In this case, it is vital that a local gap interval is determined by each access point and then applied in a way that ensures subsequent traffic bursts received by the central network server from access points are desynchronized as much as possible. If not, then the overload condition is not likely to be stably removed as rapidly as possible without undue congestion.
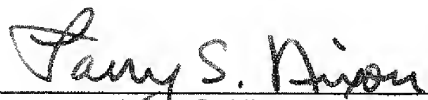
Of course, there are additional features recited in independent claims and especially in the various dependent claims that add even further patentable distinction from the cited prior art.

- 17 -

Rowland Geoffrey HUNT, *et al.*
Serial No. 10/588,726
June 29, 2011

Accordingly, this entire application is now believed to be in allowable condition, and a formal notice to that effect is earnestly solicited.

Respectfully submitted,

NIXON & VANDERHYE P.C.

By: _____
Larry S. Nixon
Reg. No. 25,640

LSN:lef

901 North Glebe Road, 11th Floor
Arlington, VA 22203-1808
Telephone: (703) 816-4000
Facsimile: (703) 816-4100

- 18 -